

IN-PLANT LOGISTICS SYSTEMS MODELING WITH SysML

Veronique Limère
Ghent University
Technologiepark 903
B-9052 Ghent-Zwijnaarde, Belgium
E-mail: Veronique.Limere@ugent.be

Leon McGinnis
Georgia Institute of Technology
765 Ferst Drive, NW
Atlanta, GA 30332, USA

Sarath Balachandran
Georgia Institute of Technology
765 Ferst Drive, NW
Atlanta, GA 30332, USA

Hendrik Van Landeghem
Ghent University
Technologiepark 903
B-9052 Ghent-Zwijnaarde, Belgium

KEYWORDS

SysML, systems modeling language, facility logistics, kitting, bulk feeding, cost model.

ABSTRACT

Up till now Systems Modeling Language (SysML) has mostly been used to model physical systems of interest. This paper shows how SysML can also be used to represent an abstract model. In this application a mathematical cost model is represented using the SysML plugin for the software MagicDraw. ParaMagic, a plugin in MagicDraw supplementary to SysML, links to Mathematica to solve the model. SysML is a formal language and offers a very intuitive graphical representation. It is therefore a useful medium to create a domain specific language for a field of knowledge. The comprehensiveness of the language, which makes it possible to incorporate specification, analysis, design, verification, and validation of systems, makes it a very valuable tool for collaboration on large projects.

INTRODUCTION

SysML is a relatively new systems modeling language which provides the syntax for very expressive models of systems. Friedenthal et al. (2008) define Systems Modeling Language (SysML) as follows:

SysML is a general purpose graphical modeling language for systems engineering applications. It is a dialect of UML™, the industry standard for modeling software-intensive systems. It supports the specification, analysis, design, verification and validation of a broad range of complex systems, including hardware, software, information, processes, personnel, and facilities.
(Friedenthal et al., 2008)

Up till now, the focus for SysML applications has been mainly on modeling physical systems of interest and not so much on modeling abstract systems and models. In this project SysML is used to graphically represent a mathematical cost model. We deal with a compact cost model for the minimization of in-plant logistics costs for feeding parts to an assembly line. To the best of our knowledge, no such application has been published before. Block definition diagrams are used to represent the structure

of the model and parametric diagrams are used to model the equations. The diagrams and their usage will be extensively discussed in two separate sections about structure and parametrics.

MagicDraw is the architecture modeling environment which is used to implement the model. SysML is packaged as a plugin to the MagicDraw tool and supports all SysML diagrams. SysML Parametrics are further supported by an additional plugin, ParaMagic, which enables computations directly from the SysML model.

PROBLEM STATEMENT AND SCOPE

Nowadays, customers put a lot of pressure on the market to obtain timely delivery and low prices. In addition, more and more variation in the product assortment is demanded and custom-made products are often requested. This current trend is explicitly perceivable in the automotive industry. Each single vehicle that comes off the line is equipped with the proper options requested by the customer. No two vehicles coming off the line consecutively are identical. This evolution towards more customization has major consequences for production organizations and their logistics department. Components do not only exist in a single variant but alternative variant parts have to be assembled. This leads to an increasing number of parts moving around on the shop floor and undoubtedly to a more complex material supply process.

In practice, different methods of material supply are practiced in the vehicle industry. Bulk feeding is the most straightforward method of line feeding. However in the automotive industry parts are often large and voluminous. Putting a full packaging unit – i.e. container or pallet or box – of each part number in the border of line would require an enormous production area and is therefore not feasible. Moreover line-operators would have to look too far to find the correct components to be assembled which would lead to a decreased productivity. To address the need for diminishing line stocks and a better organized border of line in order to facilitate the operator's task, kitting is introduced as a counterpart of bulk line feeding. To introduce the reader to the problem setting, and for the understanding of the SysML model we shortly define the two main methods of materials supply.

Bulk feeding

Feeding a line in bulk means that no parts are yet assigned to a certain end product when they are delivered to the line. No logical combinations are made between the parts and each stock keeping unit is delivered in a full container quantity.

Kitting

A *kit* is a specific collection of components and subassemblies that together (i.e. in the same container) support one or more subassembly operations for a given product or “shop order” (Bozer and McGinnis, 1992) and *kitting* is the practice of putting together a kit of components and/or subassemblies – according to a future assembly schedule – before delivery to the shop floor. Kit assembly takes place in a separate supermarket area.

Because of the increasing number of parts moving around on the shop floor and the corresponding large number of transactions, considerable amounts of money are involved. Therefore, performing parts handling activities efficiently and assigning the most appropriate line feeding methods to the parts at hand is of major importance. In this viewpoint a cost model can be used to look for an optimal assignment solution.

In this paper a compact cost model is considered. The focus is not on representing all the features of the line feeding problem, but instead on illustrating how SysML can be used to model an optimization model graphically and the benefits of doing this. More specifically, the model considers the influence of the materials supply method, i.e. bulk feeding or kitting, on the operator efficiency at the assembly line, and the labor costs for the operator that has to do the additional material handling when kitting is preferred are included. Optimal decision making will then be guided by the objective of minimizing all costs.

Notation

COH	Cost of an operator hour (€/h)
Vel	Velocity of an operator (m/h)
$YQTY_i$	Usage (units/year) of part number i
tp_i^b	The time (h) to pick a unit of part number i from a bulk container
tp_i^k	The time(h) to pick a unit from a kit
Δ_i^k	The distance (m) for the operator in the supermarket to pick part number i from a bulk container to kit
f_i	Percentage of end products using part number i
m_i	Number of units of part number i assembled per end product (bill of material)
B^{kit}	The batch size for assembling kits, i.e. the number of kits an operator assembles at once
tk_i	The time (h) to pick a part number i in the supermarket from its bulk container in order to kit it

A binary decision variable assigns parts to the appropriate materials supply method:

x_i	$x_i = 1$: Bulk feeding
	$x_i = 0$: Kitting

Mathematically we can represent the objective function by the following:

$$\text{Min } C_{total} = \text{Min } (C_{pick} + C_{kit})$$

$$C_{pick} = \sum_i [COH \cdot YQTY_i \cdot [x_i \cdot tp_i^b + (1 - x_i) \cdot tp_i^k]]$$

$$C_{kit} = \sum_i [COH \cdot (1 - x_i) \cdot YQTY_i \cdot tk_i]$$

With:

$$tk_i = \left(2 \cdot \frac{\Delta_i^k}{Vel} \right) / f_i \cdot m_i \cdot B^{kit}$$

C_{pick} represents the cost of labor at the assembly line. The picking time for the production operator at the line is dependent on the materials supply method. If parts are kitted, the operator productivity will be higher and the picking time shorter ($\forall i: tp_i^b \geq tp_i^k$).

C_{kit} represents the cost of labor in the kitting area. If parts are kitted, additional labor is needed to assemble the kits in advance. Productivity of picking in a kitting area is higher because kits can be assembled in batches and kitting areas can be set up to contribute to efficient picking.

MODELING OF STRUCTURE

In order to model the problem at hand we used two of the four important pillars of SysML (Figure 1), i.e. structure and parametrics. A link between the two is obtained through value binding. In this section we start by describing the structure. The next section concentrates on parametrics.

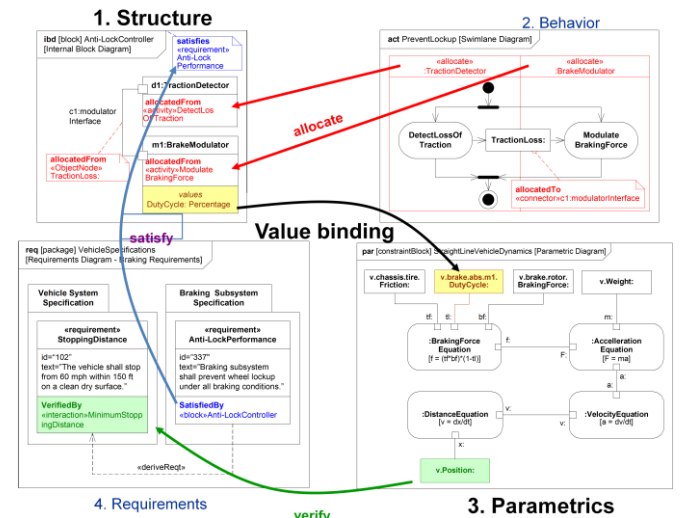


Figure 1: The Four Pillars of SysML
(Source: Friedenthal, www.omg-sysml.org)

As mentioned in the previous section, the main objective is to calculate the total cost for supplying all parts to the line. This cost can be obtained using a sequence of equations. To allow for additional complexity to be easily added in the future, we use object-oriented modeling techniques. This allows a model to be built from simple, independent, and potentially reusable subsystems, and to be tied together only at the highest levels. Obviously, this object-oriented structure will contribute to an overall comprehensible model and a clear structure. SysML supports object-oriented

modeling by use of Block definition diagrams (BDD). A *Block Definition Diagram (BDD)* describes the organization of the structure, the hierarchy of system, subsystems, and all the elements that make up the system.

Moreover, in SysML the design of complex systems is achieved in a top-down approach. We will illustrate this top-down approach for our example model starting with the composition of the overall total cost, and gradually concentrating more on details of the sub-costs.

The structure of the total cost is represented in the Block Definition Diagram ‘Totalcost’ (Figure 2). In the center three blocks are represented. A *block* is a very general modeling concept in SysML that is used to model a wide variety of entities that have structure such as systems, hardware, software, physical objects, and abstract entities (Friendenthal et al. 2008). The interconnections between the blocks are composite association paths that relate the whole to its parts. In this case abstract entities are represented, the total cost which is composed of a picking cost and a kitting cost. In the right upper corner a constraint is displayed. A *constraint block* is a special kind of block used to define equations (Friendenthal et al. 2008). The total cost equation, i.e. total cost being a sum of the picking cost and the kitting cost, is clarified in the constraint TotalcostEqn.

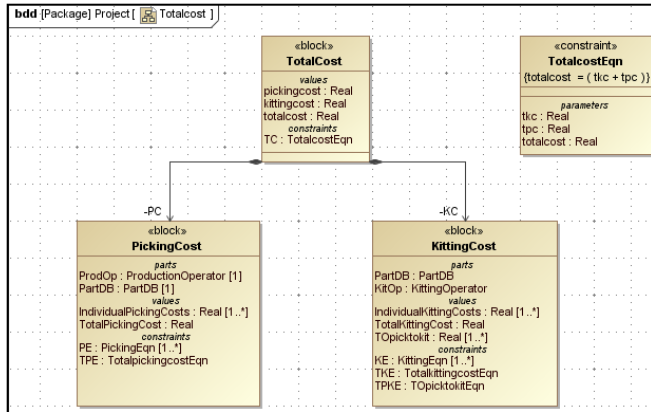


Figure 2: Block Definition Diagram – Total cost

The following Block Definition Diagrams deal with more detail of the sub-costs. PickingBDD (Figure 3) and Kitting BDD (Figure 4) structure respectively the picking cost and the kitting cost. The fact that picking cost and kitting cost are each shown in a separate Block Definition Diagram helps to maintain a clear overview.

The picking cost for the production operator at the line C_{pick} (Figure 3) is calculated from the part database and data about the production operator. The part database gives information for each of the parts that need to be supplied to the line and is structured as an array using value properties with aggregate data types. A part is characterized by its part number, its $YQTY_i$, its weight, its tp_i^b , its frequency of occurrence, its bill of material, its Δ_i^k , and its decision variable x_i . The production operator has an hourly labor cost, and a constant tp^k .

The calculation of the total picking cost is done in two steps. First, the constraint ‘PickingEqn’ calculates individual picking costs for each of the parts as an aggregate value

property. The second constraint ‘TotalpickingcostEqn’ then calculates the total picking cost as a sum of the individual picking costs, by means of the aggregate sum function ‘sum()’.

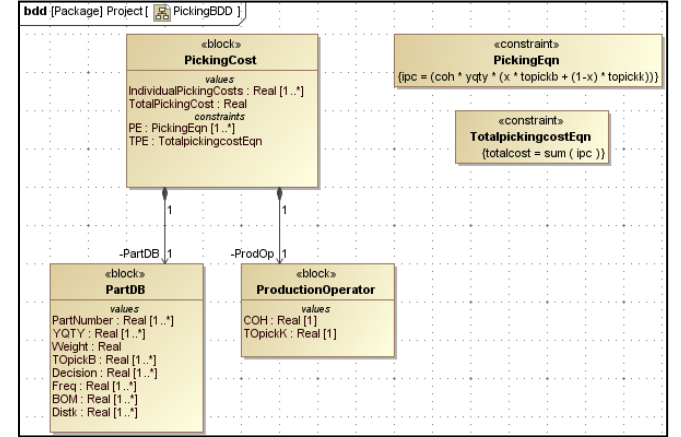


Figure 3: Block Definition Diagram – Picking cost

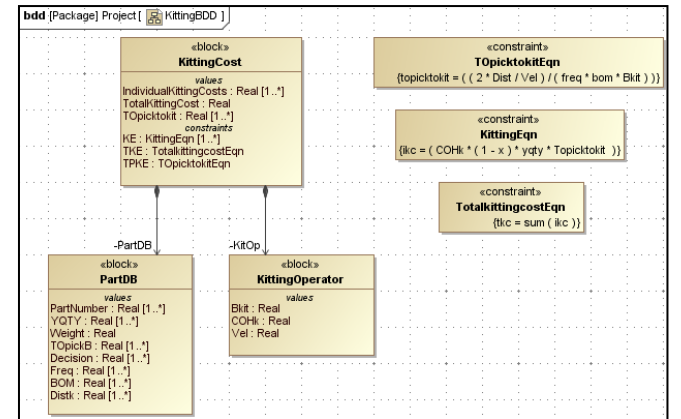


Figure 4: Block Definition Diagram – Kitting cost

The kitting cost C_{kit} (Figure 4) is modeled similarly. An extra constraint is needed for the representation of the operator time needed to pick a unit of part number i from the kitting area.

MODELING OF PARAMETRICS

With the use of Block Definition Diagrams, the structure of the model is described. Additionally Parametric Diagrams are used to relate the constraints and parameters. *Parametric diagrams* are used to create systems of equations that can constrain the properties of blocks (Friendenthal et al. 2008). For more information about Parametrics we refer to Peak et al. (2007).

For each of the constraints a Parametric Diagram is created. These Parametric Diagrams take care of the correct value binding of all parameters of the model. We will show all of the Parametric Diagrams in the same structure as the Block Definition Diagrams. The diagrams should be clear without much further explanation.

In Figure 5 at the top the two inputs for the total cost equation are linked to the constraint and below the output is displayed.

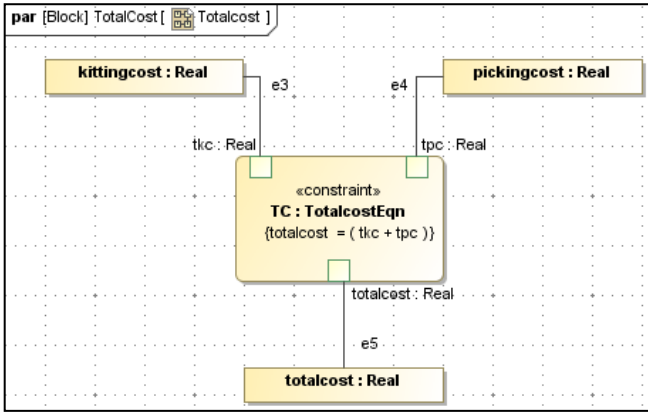


Figure 5: Parametric Diagram – Total cost

Figure 6 displays the inputs and output for individual picking costs per part in the first diagram and the transformation to a total picking cost in the second diagram.

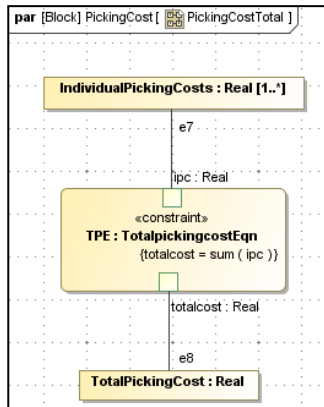
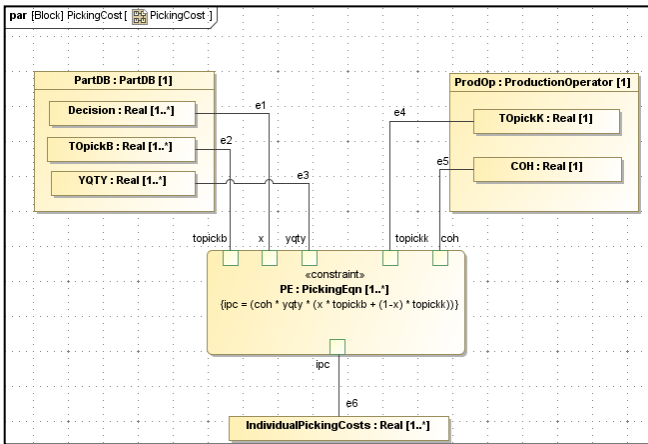


Figure 6: Parametric Diagrams – Picking cost

Figure 7 presents the same for the kitting cost. An extra first diagram is added for the representation of tk_i , an intermediate factor for the calculation of the individual kitting costs. The second diagram displays the inputs – including the previously defined tk_i – and output for individual kitting costs per part. The last diagram shows the transformation to a total kitting cost.

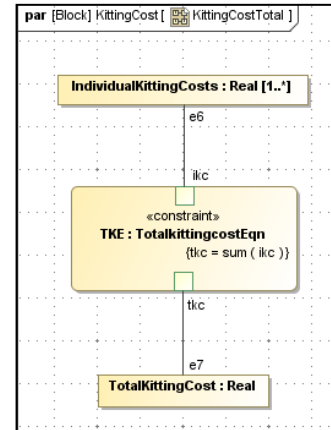
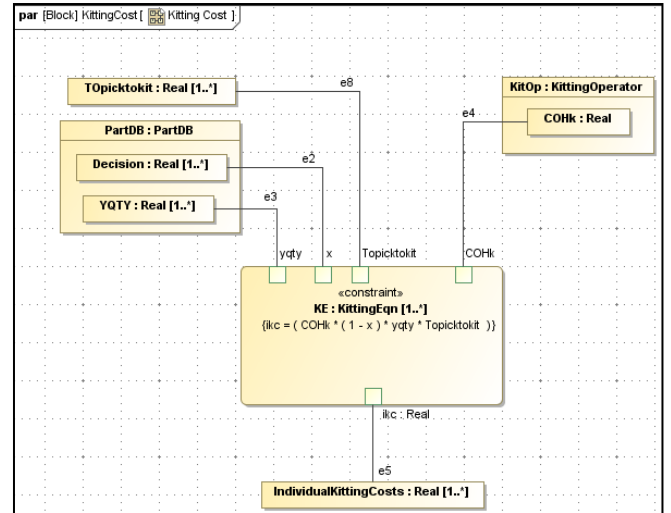
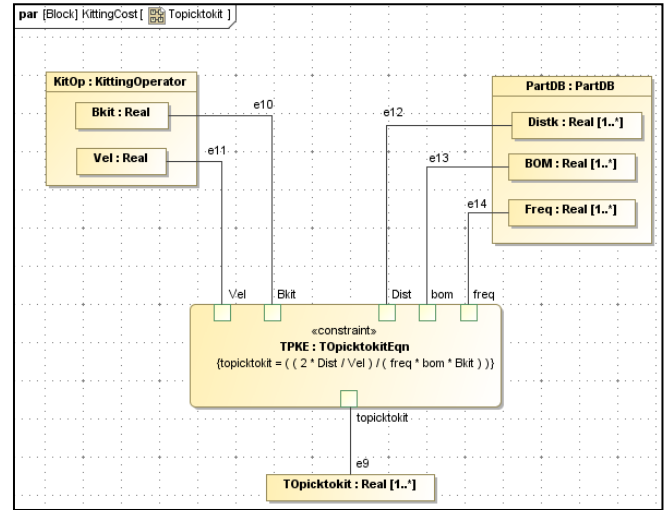


Figure 7: Parametric Diagram – Kitting cost

VALIDATION

To avoid errors, the model schema is at this point validated through the creation of a CXS_heading in MagicDraw. Bugs are revealed and removed in order to obtain a valid structure for the SysML model.

As a second tool the SysML parametrics "flattened graph" visualization tool, BuzzToys Panorama, is used to validate the model. The graphic representation (Figure 8) gives a clear overview and by walking through the tree structure on

the left it can be checked that the subsystems are linked correctly to the parameters and to each other.

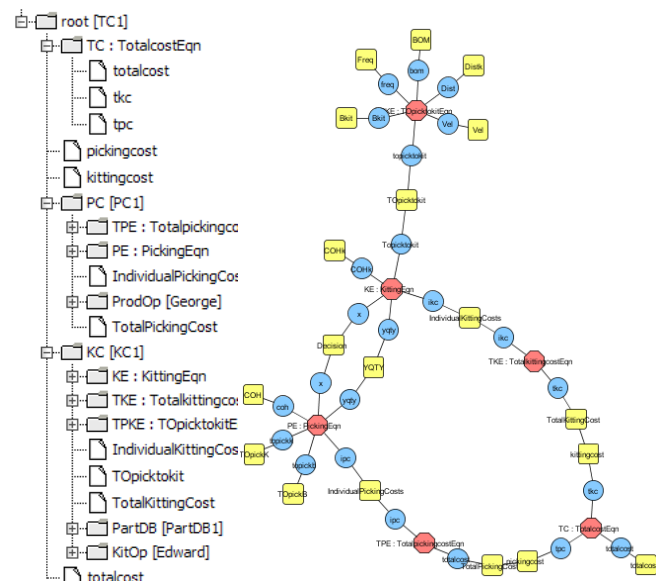


Figure 8: Panorama view – Total cost

SOLVING THE MODEL FOR A GIVEN INSTANCE

An instance is an example of the model with specific values assigned to the given parameters and which can be solved for the unknowns. In this section we will explain how the SysML model can be used to solve for total costs.

A new Block Definition Diagram is created to structure the instance. This Block Definition Diagram is populated with instance blocks and appropriate connections are made. To add values for the instance we made use of the ‘read from’ Excel functionality. A part database is entered in an Excel file which later is used to write the results. In MagicDraw, the slots to put the instance values need to be created beforehand and a causality type needs to be given. The picking cost, kitting cost and total cost variables are assigned target causalities. To avoid errors, the instance is validated through the creation of a CXI_heading. Bugs are removed from the model to obtain a valid structure.

The model is solved with ParaMagic. ParaMagic connects to the Mathematica solver which uses the equations built in the parametric diagrams to obtain the desired results. In Figure 9 the ParaMagic interface is shown. The input parameters are assigned a causality type ‘given’, and the output variables are assigned a causality type ‘target’. The ‘Solve’ command allows to connect to Mathematica and the ‘Update to SysML’ command allows to pass the results on to the visual model afterwards.

Finally the solution is written to the original Excel file by the ‘write to’ Excel functionality.

CONCLUSIONS AND FURTHER RESEARCH

The purpose of the project is to model the in-plant logistics processes of parts to an automotive assembly line and create a tool for analyzing the costs of the materials supply.

Structure and Parametric Diagrams in SysML were used for modeling and ParaMagic was used as a solver.

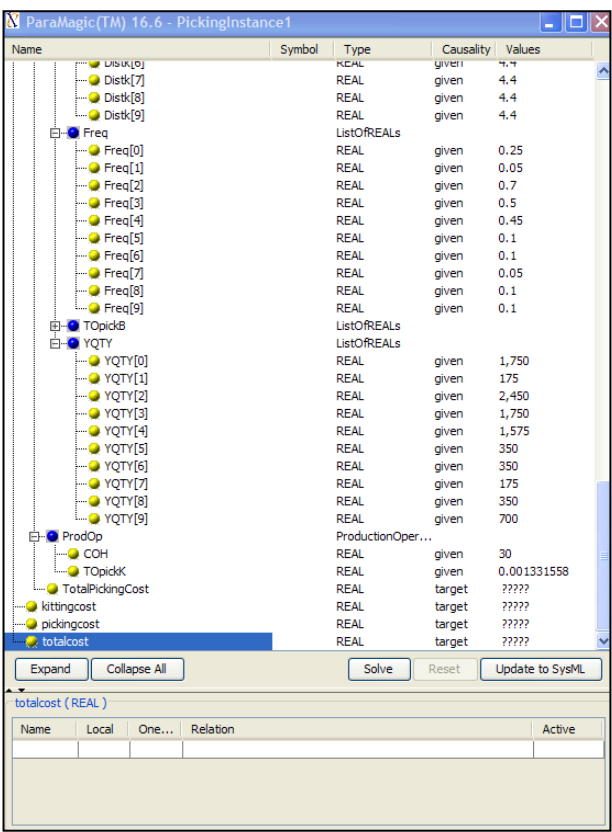


Figure 9: ParaMagic interface

Modeling in SysML has two main benefits. First, the model creates a common basis for understanding and a domain specific language is created. The structure of the model is clear and is graphically represented in a comprehensible way. Secondly, the model does not only describe the structure, which could equally be done by use of flow charts or other visual tools, but the model can be used for analysis purposes as well. This avoids duplication of efforts for building the descriptive model and then another computational model. For analysis purposes SysML supports a broad range of integration and interoperation with specific solvers, thus it enables tight integration between description and analysis. In this project ParaMagic is used to interoperate with Mathematica in order to solve the model.

The work on this project attempts to lay the groundwork for the implementation of a complete model for the problem of determining the optimal configuration of kitted and bulk fed parts to an assembly line. It is, in effect the starting point towards facilitating the use of SysML to completely specify and quantify the problem.

REFERENCES

Bozer, Y.A. and L.F. McGinnis. 1992. “Kitting versus line stocking.” *International Journal of Production Economics*, 28, 1-19.

Friedenthal, S.; A. Moore; and R. Steiner. 2008. *A Practical Guide to SysML. The Systems Modeling Language*. Elsevier Inc.

Peak, R.S. R.M. Burkhart, S.A. Friedenthal, M.W. Wilson, M. Bajaj, and I. Kim. 2007. “Simulation-Based Design Using SysML – Part 1: A Parametrics Primer.” *INCOSE Intl. Symposium, San Diego*.